

Documento de testes

Engenharia de Software

GISLENO RODRIGUES DE ALENCAR SILVA JÚNIOR - 511938, JEAN MORAIS DA SILVA - 553107, DAVID SILVA FERNANDES DE OLIVEIRA - 539402, MARCOS DA SILVA NASCIMENTO - 509773, JOÃO PEDRO HOLANDA AMORIM - 552815

Sistema de monitoramento antifurto

1. Testes Manuais.....	1
2. Testes Unitários.....	1
2.1. Testes Realizados.....	1

1. Testes Manuais

Os testes foram realizados pela equipe para validação do sistema e para correções/melhorias de funções do sistema. A seguir, segue o link de acesso para os testes para uma melhor visualização dos mesmos:

<https://docs.google.com/spreadsheets/d/19ABWGYOEA3pVMhplAdOJjzzfaIWtkUIJ/edit?usp=sharing&ouid=106424167195033794830&rtpof=true&sd=true>

2. Teste unitários

Os testes unitários foram realizados para validar as funcionalidades críticas do sistema embarcado (RFID) e do sistema web, abrangendo:

- Persistência de dados (banco de dados PostgreSQL).
- Comunicação entre camadas (embarcado → servidor → frontend).
- Regras de negócio (alertas, movimentações não autorizadas, históricos).

2.1. Testes Realizados

(Baseados nos Requisitos Funcionais RF001 a RF025)

1. Testar conexão com o banco de dados (testDatabaseConnection)

Link do teste: DatabaseTest.java

Objetivo	Verificar se a conexão com o banco de dados PostgreSQL é estabelecida corretamente.
-----------------	--

Descrição	O teste tenta conectar-se ao banco de dados utilizando JDBC e valida se a conexão não é nula.
-----------	---

Resultado	Aprovado. A conexão foi estabelecida e fechada sem erros.
-----------	---

2. Testar leitura de tag RFID (testReadRFIDTag)

Link do teste: `RFIDServiceTest.java`

Objetivo	Validar se o sistema embarcado detecta e decodifica tags RFID corretamente.
-----------------	--

Descrição	Simula a leitura de uma tag RFID com ID único e verifica se o sistema registra o evento.
-----------	--

Resultado	Aprovado. A tag foi lida e seu ID único foi registrado no log.
-----------	--

3. Testar detecção de movimento não autorizado (testUnauthorizedMovementAlert)

Link do teste: `AlertServiceTest.java`

Objetivo	Garantir que o sistema gere um alerta quando um item sai do perímetro autorizado.
-----------------	--

Descrição	Simula a movimentação de uma tag para fora da zona configurada e verifica se um alerta é disparado.
-----------	---

Resultado	Aprovado. O sistema gerou um alerta (registrado no banco de dados e enviado via e-mail mock).
-----------	---

4. Testar armazenamento local temporário (testLocalStorageOnNetworkFailure)

Link do teste: `EmbeddedSystemTest.java`

Objetivo	Verificar se os dados são armazenados localmente quando a rede falha.
-----------------	--

Descrição	Desativa a conexão de rede, simula leituras RFID e reinicia o sistema. Após reconexão, verifica se os dados são sincronizados.
Resultado	Aprovado. Os dados foram salvos localmente e sincronizados após a rede voltar.

5. Testar cadastro de item patrimonial (testAddPatrimonialItem)

Link do teste: `ItemControllerTest.java`

Objetivo **Validar o cadastro de um novo item com tag RFID no sistema web.**

Descrição Submete um formulário com nome, tipo, localização e ID da tag RFID. Verifica se o item aparece na lista.

Resultado Aprovado. O item foi persistido no banco de dados e listado na interface.

6. Testar envio de alerta por e-mail (testEmailNotification)

Link do teste: `NotificationServiceTest.java`

Objetivo **Garantir que alertas sejam enviados por e-mail aos contatos cadastrados.**

Descrição Mocka um evento de movimento não autorizado e verifica se o serviço de e-mail (SendGrid) é acionado.

Resultado Aprovado. O e-mail foi disparado para os destinatários configurados.

7. Testar filtro de histórico por data (testFilterHistoryByDate)

Link do teste: `HistoryControllerTest.java`

Objetivo **Validar a filtragem do histórico de localização por intervalo de tempo.**

Descrição	Busca registros entre duas datas e verifica se apenas os eventos no intervalo são retornados.
Resultado	Aprovado. O filtro aplicado retornou apenas os registros dentro do período especificado.

8. Testar desativação temporária de rastreador (testDisableTracker)

Link do teste: `TrackerServiceTest.java`

Objetivo	Verificar se um rastreador pode ser desativado temporariamente sem gerar alertas.
Descrição	Desativa um rastreador e simula movimentação. Válida se nenhum alerta foi gerado.
Resultado	Aprovado. O sistema ignorou as movimentações durante o período de desativação.